# Practical 5: The epigenetic clock: Predicting biological age on new data

Garyfallos Konstantinoudis

Spring Term 2025

## Part 1: The epigenetic clock: Non-parametric prediction using random forests

In this practical we consider again the same data on $n = 409$ healthy mice and methylation of $p = 3,663$ conserved methylation sites as in the last two weeks. This time we are interested in random forests and see if random forests will provide a better prediction rule than penalised regression techniques. Install and load the package

```
library("randomForest")
```

Load the dataset, that contains the methylation matrix as predictor matrix and the age of the mice (in months) stored in the vector y. Familiarise yourself with the dataset using the following commands

```
load("../data/data_epigenetic_clock_control")
#alternatively try load("../../data/data_epigenetic_clock_control.dms")
y = control_mice$y_control
x = control_mice$x_control
dim(x)
```

```
## [1]  409 3663
```

Question 1.1

First compute a random forest with the options 'ntree=100' and 'importance = TRUE' and save the output to an object call rf.out or similar.

Question 1.2

Random forests have an intrinsic way of assigning variable importance to the predictors. How do random forests with quantitative outcome rank variables according to their importance? Rank the variables according to their importance and display the 10 most important methylation sites for ageing in the random forest algorithm. Use the varImpPlot() function to visualise the variable importance.

Question 1.3

How would you run bagging using the randomForest function? What is the interpretation of the mtry option?

Question 1.4

Next step is to write a prediction function for the random forest algorithm. Re-use your code from last week (Practical 3 Question 2) and see the prediction function for a linear regression model below. Set the number of trees as an open parameter.

```
predfun.lm = function(train.x, train.y, test.x, test.y){
    #fit the model and build a prediction rule
    lm.fit = lm(train.y ~ ., data=train.x)
    #predict the new observation based on the test data and the prediction rule
    ynew = predict(lm.fit, test.x )
    #compute mse as squared difference between predicted and observed outcome
    out = mean( (ynew - test.y)^2 )
    return( out )
}
```

Question 1.5

Compare the prediction performance using ntree = 10 and ntree = 100 random trees to train the random forest. To this end we use the

```
library(crossval)
```

package and compare the two parameters using $k$-fold cross-validation (cv) using $k = 5$ folds.

Question 1.6

Finally, we want to know if random forests outperform regularised regression with respect to prediction performance. Re-use the following code from last week (Practical 3 Question 2) as prediction rule for glmnet()

```
library(glmnet)
predfun.glmnet = function(train.x, train.y, test.x, test.y, lambda = lambda, alpha=alpha){

    #fit glmnet prediction rule
    glmnet.fit = glmnet(x=train.x, y=train.y, lambda = lambda, alpha=alpha)
    #predict the new observation based on the test data and the prediction rule
    ynew = predict(glmnet.fit , newx=test.x)
    # compute squared error risk (MSE)
    out = mean( (ynew - test.y)^2 )
    return( out )
}
```

and perform cv to tune the regularisation parameter.

```
library(foreach)
set.seed(12)
lasso.cv = cv.glmnet(x,y,family="gaussian",alpha=1, type.measure="mse")
set.seed(123)
ridge.cv = cv.glmnet(x,y,family="gaussian",alpha=0, type.measure="mse")
set.seed(1234)
a = seq(0.05, 0.95, 0.05)
search = foreach(i = a, .combine = rbind)%do%{
            cv = cv.glmnet(x,y,family = "gaussian", type.measure = "mse", alpha = i)
            data.frame(cvm = cv$cvm[cv$lambda == cv$lambda.1se],
             lambda.1se = cv$lambda.1se, alpha = i)
}
elasticnet.cv = search[search$cvm == min(search$cvm), ]
```

Finally, use cv as implemented in crossval() to determine if random forests (ntree = 100) is better than lasso and elastic net as implemented in the glmnet package.

## Part 2: The epigenetic clock: Evaluating the impact of nitrogen dioxide on biological age

An environmental research group gets in contact with us. They have measured the methylation profile of $n = 131$ mice of which 36 mice have been exposed to nitrogen dioxide (NO2) for 10 hours a day for 10 weeks. The remaining 95 mice are healthy controls. Load the data and familiarise yourself with the data structure.

```
load("../data/data_epigenetic_clock_experimental")
#alternatively try load("../../data/data_epigenetic_clock_experimental.dms")
exposed = experimental_mice$exposed
table(exposed)
```

```
## exposed
##  0  1
## 95 36
```

```
x.test = experimental_mice$x_exp
dim(x.test)
```

```
## [1]  131 3663
```

The working hypothesis is that NO2 exposure reduces the biological age of mice and makes the exposed mice age more quickly. The environmental research group asks us to predict the biological age of the mice using our algorithms to determine the biological age of a mouse.

Question 2.1

Predict the biological age of the $n = 131$ new mice using the lasso model as implemented in the glmnet package. Use again $lambda.1se as the regularisation parameter (as computed in Question 1.6) and save the predicted age in an object called lasso.hat.

Question 2.2

Predict the biological age of the $n = 131$ new mice using the ridge model. Use again $lambda.1se as the regularisation parameter (as computed in Question 1.6) and save the predicted age in an object called ridge.hat.

Question 2.3

Predict the biological age of the $n = 131$ new mice using the elastic net model. Use again $lambda.1se as the regularisation parameter (as computed in Question 1.6) and save the predicted age in an object called enet.hat.

Question 2.4

Predict the biological age of the $n = 131$ new mice using the a random forest model. This time use 'ntree=500' to build a reliable prediction rule and save the predicted age in an object called rf.hat.

Question 2.5

Can you confirm the working hypothesis that NO2 exposure reduces the biological age of mice? Perform a $t$-test to see if the predicted biological age of the mice differs significantly between exposed and healthy mice. Perform a $t-$test for all four predictions done in Question 2.1-2.4. How will you advice the environmental research group?

## Part 3: Decision trees and random forests: Survival on the Titanic

The sinking of the titanic was one of the greatest disaster in navel history. After colliding with an iceberg, the titanic sank and 1,502 out of 2,224 passengers and crew were killed. The following data set has collected information on n=1,309 of the passengers and their survival.

```
titanic = read.csv("../data/titanic.csv")
dim(titanic)
```

```
## [1] 1309   14
```

```
table(titanic$survived)
```

```
##
##   0   1
## 809 500
```

Here we use decision trees and random forest to analyse the titanic data. Make sure to have the following two packages

```
library(tree)
library(randomForest)
```

installed.

Question 3.1

Fit a decision tree on the titanic data using the following predictor matrix including passenger class, sex, age, number of siblings/spouses aboard, and number of parents/children aboard after excluding missing values.

```
x=cbind(titanic$pclass, titanic$sex, titanic$age, titanic$sibsp, titanic$parch)
rm = which(is.na(titanic$age)==TRUE)
x.input = x[-rm,]
dim(x.input)
```

```
## [1] 1046    5
```

```
colnames(x.input) = c("pclass", "sex", "age", "sibsp", "parch")
y.input = as.factor(titanic$survived[-rm])
table(y.input)
```

```
## y.input
##   0   1
## 619 427
```

Use the function tree in the tree package.

Question 3.2

What is a concern when fitting a single decision tree?

Question 3.3

Prune your tree using cross-validation (cv.tree) and use the option FUN = prune.misclass for the misclassification rate as criterion. Choose the model with the lowest misclassification error and plot the tree. How do you interpret the decision tree?

Question 3.4

Finally fit a random forest to the data and look at the variable importance. What was the key variable for survival in the titanic disaster?

## Part 4 (optional): Spotify data: Which song features predict if a song is likely to be skipped?

We look again at the spotify data from practical 3

```
load("../data/spotify.Rdata")
```

and consider the binary outcome data if a song is likely to be skipped

```
y_bin = spotify.data$y_bin
table(y_bin)
```

```
## y_bin
## FALSE  TRUE
##   853  3466
```

As predictors we consider the variables given in the x_mat matrix. For more information on the features, please see https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/.

```
x_mat = spotify.data$x_mat
colnames(x_mat)
```

```
##  [1] "age"                   "duration"              "us_popularity_estimate"
##  [4] "acousticness"          "beat_strength"         "danceability"
##  [7] "dyn_range_mean"        "energy"                "flatness"
## [10] "instrumentalness"      "liveness"              "loudness"
## [13] "mechanism"             "speechiness"           "tempo"
## [16] "valence"
```

Question 4.1

Fit a random forest model that uses x_mat as predictors. Which song features are important to predict if a song is likely to be skipped?

Question 4.2

Contrast these findings with the results from the elastic net fit in practical 3 Question 4.4. How would you interpret the different findings?